Isoperimetric Graph Partitioning for Image Segmentation

Leo Grady and Eric L. Schwartz

Abstract

Spectral graph partitioning provides a powerful approach to image segmentation. We introduce an alternate idea that finds partitions with a small isoperimetric constant, requiring solution to a linear system rather than an eigenvector problem. This approach produces the high quality segmentations of spectral methods, but with improved speed and stability.

Index Terms

Graph-theoretic methods, graphs and networks, graph algorithms, image representation, special architectures, algorithms, computer vision, applications.

I. Introduction: The Isoperimetric Problem

GRAPH partitioning has been strongly influenced by properties of a combinatorial formulation of the classic isoperimetric problem: *For a fixed area, find the region with minimum perimeter.*

Define the **isoperimetric constant** h of a manifold as [1]

$$h = \inf_S \frac{|\partial S|}{\mathrm{Vol}_S}, \tag{1}$$

where S is a region in the manifold, Vol_S denotes the volume of region S, $|\partial S|$ is the area of the boundary of region S, and h is the infimum of the ratio over all possible S. For a compact manifold, $\mathrm{Vol}_S \leq \frac{1}{2}\mathrm{Vol}_{\mathrm{Total}}$, and for a noncompact manifold, $\mathrm{Vol}_S < \infty$ (see [2], [3]).

We show in this paper that the set (and its complement) for which h takes a minimum value defines a good heuristic for image segmentation. In other words, finding a region of an image that is both large (i.e., high volume) and that shares a small perimeter with its surroundings (i.e., small boundary) is intuitively appealing as a "good" image segment.

Previous applications of graph partitioning to image segmentation have yielded successful algorithms. In general, previous approaches have been based on spectral graph theory [4], [5] and on max-flow/min cut

L. Grady is with Siemens Corporate Research — Department of Imaging and Visualization, Princeton, NJ 08540, E-mail: leo.grady@siemens.com

E. L. Schwartz is with the Departments of Cognitive and Neural Systems and Electrical and Computer Engineering, Boston University, Boston, MA 02215, E-mail: eric@bu.edu

[6], [7] methods. Although the isoperimetric criterion is intuitively similar to the partitioning criterion used in the spectral graph theory, the seemingly slight difference in formulation allows solution by a system of linear equations rather than an eigenvector problem. A system of linear equations is desirable because it improves both speed and stability. Furthermore, there are situations for which spectral approaches have degenerate solutions. The minimum-cut approaches to segmentation [6], [7] tend to produce small partitions or to require a large number of (generally user-specified) sinks/sources. The isoperimetric algorithm favors larger partitions, and thus avoids this problem.

II. The Isoperimetric Partitioning Algorithm

A **graph** is a pair $G = (V, E)$ with vertices (nodes) $v \in V$ and edges $e \in E \subseteq V \times V$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. Let $n = |V|$ and $m = |E|$ where $|\cdot|$ denotes cardinality. A **weighted graph** has a value (typically nonnegative and real) assigned to each edge called a **weight**. The weight of edge e_{ij}, is denoted by $w(e_{ij})$ or w_{ij}. Since weighted graphs are more general than unweighted graphs (i.e., $w(e_{ij}) = 1$ for all $e_{ij} \in E$ in the unweighted case), we will develop all our results for weighted graphs. The **degree** of a vertex v_i, denoted d_i, is $d_i = \sum_{e_{ij}} w(e_{ij}) \;\; \forall \, e_{ij} \in E$.

For a graph, G, the **isoperimetric constant** [2], h_G is

$$h_G = \inf_S \frac{|\partial S|}{\mathrm{Vol}_S}, \tag{2}$$

where $S \subset V$ and $\mathrm{Vol}_S \leq \frac{1}{2}\mathrm{Vol}_V$. In graphs with a finite node set, the infimum in (2) is a minimum. Since the present context is that of finite graphs, we will henceforth use the minimum in place of the infimum. The boundary of a set, S, is defined as $\partial S = \{e_{ij} | v_i \in S, v_j \in \overline{S}\}$, where $\overline{S}$ denotes the set complement, and

$$|\partial S| = \sum_{e_{ij} \in \partial S} w(e_{ij}). \tag{3}$$

In order to determine a notion of volume for a graph, a metric must be defined. Different choices of a metric lead to different definitions of volume and even different definitions of a combinatorial Laplacian operator (see [3], [8]). Dodziuk suggested [9], [10] two different notions of combinatorial volume,

$$\mathrm{Vol}_S = |S|, \tag{4}$$

and

$$\mathrm{Vol}_S = \sum_i d_i \;\; \forall \, v_i \in S. \tag{5}$$

One may view the difference between the definition of volume in (4) and that in (5) as the difference between what Shi and Malik term the "Average Cut" versus their "Normalized Cut" [4], although the isoperimetric ratio (with either definition of volume) corresponds to neither criterion. Traditional spectral partitioning [11] employs the same algorithm as Ncuts, except that it uses the combinatorial Laplacian defined by the metric associated with (4). In agreement with [4], we find that the second metric (and hence, volume definition) is more suited to image segmentation since regions of uniform intensity are given preference over regions that simply possess a large number of pixels. Therefore, we will use Dodziuk's second metric definition and employ volume as defined in (5).

For a given set, S, we term the ratio of its boundary to its volume the **isoperimetric ratio**, denoted by $h(S)$ (i.e., the argument of $\inf$ in (2)). The **isoperimetric sets** for a graph, G, are any sets S and $\overline{S}$ for which $h(S) = h_G$ (note that the isoperimetric sets may not be unique for a given graph). The specification of a set satisfying the volume constraint, together with its complement may be considered as a *partition* and therefore the term is used interchangeably with the specification of a set satisfying the volume constraint. Throughout this paper, a good partition is considered to be one with a low isoperimetric ratio (i.e., the optimal partition is represented by the isoperimetric sets themselves). Therefore, the goal is to maximize Vol_S while minimizing $|\partial S|$. Unfortunately, finding isoperimetric sets is an NP-hard problem [2]. The algorithm of this paper may be considered to be a heuristic for finding a set with a low isoperimetric ratio that runs in low-order polynomial time.

A. Derivation of Isoperimetric Algorithm

Define an indicator vector, x, that takes a binary value at each node

$$x_i = \begin{cases} 0 & \text{if } v_i \in \overline{S}, \\ 1 & \text{if } v_i \in S. \end{cases} \tag{6}$$

Note that a specification of x may be considered a partition.

Define the $n \times n$ matrix, L, of a graph as

$$L_{v_i v_j} = \begin{cases} d_i & \text{if } i = j, \\ -w(e_{ij}) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The notation $L_{v_i v_j}$ is used to indicate that the matrix L is being indexed by vertices v_i and v_j. This matrix is also known as the **admittance matrix** in the context of circuit theory or the **Laplacian matrix** in the context of finite difference methods (and in the context of [9]).

By definition of L,

$$|\partial S| = x^T L x, \tag{8}$$

and $\mathrm{Vol}_S = x^T d$, where d is the vector of node degrees. If r indicates the vector of all ones, minimizing (8) subject to the constraint that the set, S, has fixed volume may be accomplished by asserting

$$\mathrm{Vol}_S = x^T d = k, \tag{9}$$

where $0 < k < \frac{1}{2}r^T d$ is an arbitrary constant and r represents the vector of all ones. We shall see that the choice of k becomes irrelevant to the final formulation. Thus, the isoperimetric constant (2) of a graph, G, may be rewritten in terms of the indicator vector as

$$h_G = \min_x \frac{x^T L x}{x^T d}, \tag{10}$$

subject to the constraint of (9). Given an indicator vector, x, $h(x)$ represents the isoperimetric ratio associated with the partition specified by x. Note that the ratio given by (10) is different from both the "ratio cut" of [12], [13] and the "average cut" of [4]. Although the criterion in (10) rewards qualitatively similar partitions to the normalized cut, average cut and ratio cut (i.e., large segments with small boundaries), what appears as a minor difference in the formulation allows us to use a solution to a system of linear equations instead of solving an eigenvector problem. Note that the ratio cut technique of [12], [13] is distinct (in algorithm and pertinent ratio) from the ratio cut of [14], which applies only to planar graphs. The advantages of solving a system of linear equations, rather than an eigenvector problem, will be discussed below.

The constrained optimization of the isoperimetric ratio is made into a free variation via the introduction of a Lagrange multiplier Λ and relaxation of the binary definition of x to take nonnegative real values by minimizing the cost function

$$Q(x) = x^T L x - \Lambda(x^T d - k). \tag{11}$$

Since L is positive semi-definite (see, [15], [16]) and $x^T d$ is nonnegative, $Q(x)$ will be at a minimum for any critical point. Differentiating $Q(x)$ with respect to x and setting to a minimum yields

$$2Lx = \Lambda d. \tag{12}$$

Thus, the problem of finding the x that minimizes $Q(x)$ (minimal partition) reduces to solving a linear system. Henceforth, the scalar multiplier 2 and the scalar Λ are dropped, since only the relative values of the solution are significant.

Unfortunately, the matrix L is singular: all rows and columns sum to zero (i.e., the vector r spans its nullspace), so finding a unique solution to (12) requires an additional constraint.

We assume that the graph is connected, since the optimal partitions are clearly each connected component if the graph is disconnected (i.e., $h(x) = h_G = 0$). Note that in general, a graph with c connected components will correspond to a matrix L with rank $(n - c)$ [15]. If we arbitrarily designate a node, v_g, to include in S (i.e., fix $x_g = 0$), this is reflected in (12) by removing the gth row and column of L, denoted by L_0, and the gth row of x and d, denoted by x_0 and d_0, such that

$$L_0 x_0 = d_0, \tag{13}$$

which is a nonsingular system of equations.

Solving (13) for x_0 yields a real-valued solution that may be converted into a partition by setting a threshold (see below for a discussion of different methods). In order to generate a segmentation with more than two parts, the algorithm may be recursively applied to each partition separately, generating subpartitions and stopping the recursion if the isoperimetric ratio of the cut fails to meet a predetermined threshold. We term this predetermined threshold the `stop` parameter and note that since $0 \leq h(x) \leq 1$, the `stop` parameter should be in the interval $(0, 1)$. Since lower values of $h(x)$ correspond to more desirable partitions, a stringent value for the `stop` parameter is small, while a large value permits lower quality partitions (as measured by the isoperimetric ratio). It was proved in [17] that the partition containing the node corresponding to the removed row and column of L must be connected, for any chosen threshold i.e., the nodes corresponding to x_0 values less than the chosen threshold form a connected component.

B. Physical analogy

Equation (12) occurs in circuit theory when solving for the electrical potentials of an ungrounded circuit in the presence of current sources [18]. After grounding a node in the circuit (i.e., fixing its potential to zero), determination of the remaining potentials requires a solution of (13). Therefore, we refer to the node, v_g, for which we set $x_g = 0$ as the **ground node**. Likewise, the solution, x_i, obtained from (13) at node v_i, will be referred to as the **potential** for node v_i. The need for fixing an $x_g = 0$ to constrain (12) may be seen not only from the necessity of grounding a circuit powered only by current sources in order to find unique potentials, but also from the need to provide a boundary condition in order to find a solution to Poisson's equation, of which (12) is a combinatorial analog. In the present case, the "boundary condition" is that the grounded node is fixed to zero.

With this interpretation of the notation used above, the three fundamental equations of circuit theory (Kirchhoff's current and voltage law and Ohm's law) may be written for a grounded circuit as

$$A_0^T y = f \quad \text{(Kirchhoff's Current Law)}, \tag{14}$$

$$Cp = y \quad \text{(Ohm's Law)}, \tag{15}$$

$$p = A_0 x \quad \text{(Kirchhoff's Voltage Law)}, \tag{16}$$

for a vector of branch currents, y, current sources, f, and potential drops (voltages), p. Note that there are no voltage sources present in this formulation. These three equations may be combined into the linear system

$$A_0^T C A_0 x = L_0 x = f, \tag{17}$$

since $A^T C A = L$ [15].

There is a deep connection between electric circuits and random walks on graphs [19], [20], which suggests the analysis of this algorithm in terms of a random walk on a graph. The electric potential calculated above for each node admits interpretation as the expected number of steps a random walker starting from that node would take in order to reach the ground, if his probability of walking from node v_i to v_j is equal to $\frac{w_{ij}}{d_i}$. In this interpretation, the threshold is in units of expected steps of a random walker to ground, chosen to partition the graph into subsets possessing the smallest isoperimetric ratio (see [21] for justification of this interpretation).

C. Algorithmic details

1) Summary of the algorithm: Applying the isoperimetric algorithm to image segmentation may be described in the following steps:

1) Find weights for all edges using (18) and build the L matrix (7).
2) Choose the node of largest degree as the ground node, v_g, and determine L_0 and d_0 by eliminating the row/column corresponding to v_g.
3) Solve (13) for x_0.
4) Threshold the potentials x at the value that gives partitions corresponding to the lowest isoperimetric ratio.
5) Continue recursion on each segment until the isoperimetric ratio of the subpartitions is larger than the `stop` parameter.

2) Choosing edge weights: In order to apply the isoperimetric algorithm to partition a graph, the image values must be encoded on the graph via edge weights. We employ the standard [4], [7] weighting function

$$w_{ij} = \exp\left(-\beta(I_i - I_j)^2\right), \tag{18}$$

where β represents a free parameter and I_i indicates the intensity value at node v_i. Note that $(I_i - I_j)^2$ may be replaced by the squared norm of a Euclidean distance in the case of vector valued data. In order to make one choice of β applicable to a wide range of data sets, we have found it helpful to normalize the intensity differences for an image before applying (18).

3) Choosing Partitions from the Solution: The binary definition of x was extended to the real numbers in order to solve (13). Therefore, in order to convert the solution, x, to a partition, a subsequent step must be applied (as with spectral partitioning). Conversion of a potential vector to a partition may be accomplished using a threshold. A **cut value** is a value, α, such that $S = \{v_i | x_i \leq \alpha\}$ and $\overline{S} = \{v_i | x_i > \alpha\}$. The partitioning of S and $\overline{S}$ in this way may be referred to as a **cut**. This thresholding operation creates a partition from the potential vector, x. Note that since a connected graph corresponds to an L_0 that is an M-matrix [16], and is therefore monotone, $L_0^{-1} \geq 0$. This result then implies that $x_0 = L_0^{-1} d_0 \geq 0$.

Employing the terminology of [22], the standard approaches to cutting the indicator vector in spectral partitioning are to cut based on the median value (the **median cut**) or to choose a threshold such that the resulting partitions have the lowest available isoperimetric ratio (the **ratio cut**). Note that in the remainder of this paper, the “ratio cut” is used in the sense of [22] (which describes a method for

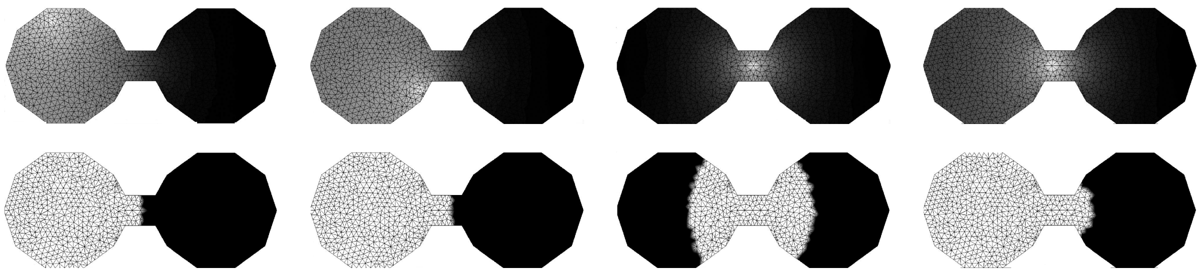


Fig. 1. An example of the effects on the solution with different choices of ground node for a problem with a known optimal partition. The top row shows the potential function (brightest point is ground) for several choices of ground while the bottom row shows the thresholded partitions. Different ground choices within an object (e.g., a ball in the dumbbell) have little effect on the partitioning/segmentation. Only a pathological choice (grounding on the optimal boundary) produces a significantly different partition. Uniform weights were employed in this example.

binarizing a real-valued solution) and not in the sense of [12], [13] or [14] (which describe complete partitioning/segmentation algorithms). The ratio cut method will clearly produce partitions with a lower isoperimetric ratio than the median cut. Unfortunately, because of the required sorting of x, the ratio cut method requires $\mathcal{O}(n\log(n))$ operations (assuming a bounded degree). The median cut method runs in $\mathcal{O}(n)$ time, but forces the algorithm to produce equal sized partitions, even if a better segmentation exists. Despite the required sorting operation for the ratio cut, the operation is still very inexpensive relative to the solution of (13) for the range of n we focus on (typically 128×128 to 512×512 images). Therefore, we have chosen to employ the ratio cut method.

4) Ground node: It will be demonstrated below that, in the image processing context, the ground node may be viewed from an attentional standpoint. However, in the more general graph partitioning context it remains unclear how to choose the ground. Anderson and Morley [23] proved that the spectral radius of L, $\rho(L)$, satisfies $\rho(L) \leq 2d_{\max}$, suggesting that grounding the node of highest degree may have the most beneficial effect on the conditioning of (13). Empirically, we have found that as long as the ground is not along the ideal cut, a partition with low isoperimetric ratio is produced.

Figure 1 illustrates this principle using the dumbbell shape Cheeger's seminal paper [1] on the relationship of the isoperimetric constant and the eigenvalues of the Laplacian on continuous manifolds. The top row shows the potentials, x, solved for using (13). The brightest node on the graph represents the ground node. For the rest of the nodes, bright nodes are closer to ground (i.e., have lower potentials) and dark nodes are further from ground. The bottom row shows the post-threshold function, i.e., after the ratio cut has been employed. The two left columns indicate a random selection of ground nodes and the two right columns represent pathological choices of ground nodes. Of the two pathological cases, the third column uses a ground in the exact center of the neck, while the last column uses a ground displaced by one node from the center. Although the grounding in the exact center produces a partition that does not resemble the known ideal partition, grounding a node displaced by one from the center (fourth column) produces a partition that is nearly the same as the ideal. This experiment illustrates that the solution is largely independent of the choice of ground node, except in the pathological case where the ground lies on the optimal cut. Moreover, it is clear that choosing a ground node in the interior of the balls is better than choosing a point on the neck, which corresponds in some sense to the above rule of choosing the point with maximum degree since a node of high degree will be in the "interior" of a region, or in an area of uniform intensity in the context of image processing.

5) Solving the System of Equations: Solving (13) is the computational core of the algorithm. It requires the solution to a large, sparse system of symmetric equations where the number of nonzero entries in L will equal $2m$.

Of the many methods for solving systems of linear equations [24], the method of conjugate gradients appears favored in this application, because it is memory efficient, parallelizable and allows a flexible trade-off of speed for accuracy. The numerical basis of the present work is the sparse matrix package provided by MATLAB [25].

6) Time Complexity: Running time depends mainly on the solution to (13). A sparse matrix-vector operation depends on the number of nonzero values, which is, in this case, $\mathcal{O}(m)$. Assuming that a constant number of iterations is required for the convergence of the conjugate gradients method, the time complexity of solving (13) is $\mathcal{O}(m)$. The ratio cut requires a $\mathcal{O}(n\log(n))$ sort. Combined, the time complexity is $\mathcal{O}(m + n\log n)$. In cases of graphs with bounded degree, then $m \leq nd_{\max}$ and the time complexity reduces to $\mathcal{O}(n\log(n))$. If a constant recursion depth may be assumed (i.e., a consistent number of "objects" in the scene), the time complexity is unchanged.

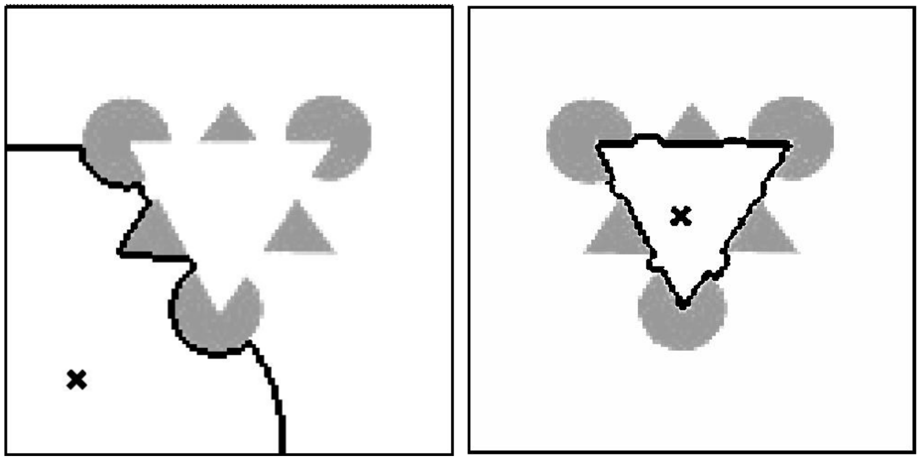
D. Relationship to Spectral Partitioning

Building on the early work of Fiedler [26], Alon [27] and Cheeger [1], who demonstrated the relationship between the second smallest eigenvalue of the Laplacian matrix (the **Fiedler value**) for a graph and its isoperimetric constant, spectral partitioning was one of the first successful graph partitioning algorithms [11]. The algorithm partitions a graph by finding the eigenvector corresponding to the Fiedler value, termed the **Fiedler vector**, and cutting the graph based on the value in the Fiedler vector associated with each node. Like isoperimetric partitioning, the output of the spectral partitioning algorithm is a set of values assigned to each node, which require cutting in order to generate partitions.

Spectral partitioning may be used [11] to minimize the isoperimetric ratio of a partition by solving

$$Lz = \lambda z, \tag{19}$$

with L defined as above and λ representing the Fiedler value. Since the vector of all ones, r, is an eigenvector corresponding to the smallest eigenvalue (zero) of L, the goal is to find the eigenvector associated with the second smallest eigenvalue of L. Requiring $z^T r = 0$ and $z^T z = n$ may be viewed as additional constraints employed in the derivation of spectral partitioning to circumvent the singularity of L (see, [28] for an explicit formulation of spectral partitioning from this viewpoint). Therefore, one way



(a) Ground in corner (b) Ground in triangle

Fig. 2. The Kaniza triangle illusion with the single bipartition outlined in black and the ground node marked with an 'x'. The ground location behaves like an attentional point in determining the segmentation. Additionally, this example demonstrates that the algorithm is capable of finding low-contrast (or zero-contrast) boundaries based on the global structure of the image without any explicit modeling of Gestalt properties.

of viewing the difference between the isoperimetric and the spectral methods is in terms of the choice of an additional constraint that allows one to circumvent the singular nature of the Laplacian L.

A second difference is that the isoperimetric method requires the solution to a sparse linear system rather than the solution to the eigenvector problem required by spectral methods of image segmentation [4], [5]. The Lanczos algorithm provides an excellent method for approximating the eigenvectors corresponding to the smallest or largest eigenvalues of a matrix with a time complexity comparable to the conjugate gradient method of solving a sparse system of linear equations [24]. However, solution to the eigenvector problem is less stable to minor perturbations of the matrix than the solution to a system of linear equations, if the desired eigenvector corresponds to an eigenvalue that is very close to other eigenvalues (see, [24]). In fact, the eigenvector problem is degenerate for graphs in which the Fiedler value has algebraic multiplicity greater than one, allowing the Lanczos algorithm to converge to any vector in the subspace spanned by the Fiedler *vectors* (if it converges at all). A square lattice with uniform weights is an example of a graph for which the Fiedler value has algebraic multiplicity greater than unity. More problematic is that even a single pixel that is nearly disconnected from its neighbors (e.g., due to high contrast) will drive another eigenvalue toward zero. Therefore, even a few nearly disconnected pixels (e.g., a black pixel surrounded by white pixels) can result in the numerical instabilities described above. This problem will be demonstrated empirically in Section III-B for noisy images. In contrast, the isoperimetric algorithm requires the solution of a linear system, and is therefore robust to the previous criticism. A formal comparison of the sensitivities of the isoperimetric, spectral and Ncuts algorithms is performed in [17].

III. Applications

A. Methods of image segmentation

It is not clear, *a priori*, how to impose a graph structure on an image. Since pixels define the discrete input, a simple choice for nodes is the pixels themselves. Traditional neighborhood connectivity employs a 4-connected or 8-connected topology. Another approach, taken by Shi and Malik [4] is to use a fully connected neighborhood within a parameterized radius of each node. We chose to use a minimal 4-connected topology since the matrix L becomes less sparse as more edges are added to the graph, and a graph with more edges requires more time to solve (13). In other work we show that that the use of more exotic neighborhoods may enhance the algorithm performance [29]. Edge weights were generated from intensity values in the case of a grayscale image or from RGB color values in the case of a color image using (18).

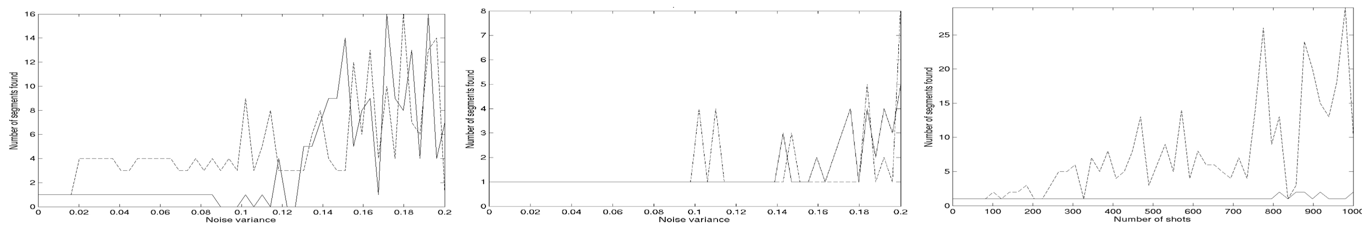
The isoperimetric algorithm is controlled by only two parameters: the parameter β of (18) and the `stop` parameter used to end the recursion. The β parameter affects how sensitive the algorithm is to changes in feature space (e.g., RGB, intensity), while the `stop` parameter determines the maximum acceptable isoperimetric ratio a partition must generate in order to accept it and continue the recursion.

Study of the classic Kaniza illusion [30] suggests that humans segment objects based on something beyond perfectly connected edge elements. The isoperimetric algorithm was used to segment the image in Figure 2, using only one level of recursion with all nodes corresponding to the black “inducers” removed. In this case, choice of the ground node is important for determining the single bipartition. If the ground node is chosen inside the illusory triangle, the resulting partition is the illusory triangle. However, if the ground is chosen outside, the triangle partition is not produced, but instead a partition that hugs the corner in which the ground is located. In this way, the ground node may be considered as representing something like an “attentional” point, since it induces a partition that favors the region of the ground node. However, note that these partitions are compatible with each other, suggesting that the choice of ground may affect only the order in which partitions are found. We believe that the ability to “complete” an object boundary is an important quality for a segmentation algorithm, since natural images frequently contain weak object boundaries.

B. Stability

Stability[1] of the solutions obtained for the isoperimetric and the spectral algorithms differs considerably. This difference is expected from perturbation analysis applied to the solutions of a linear system versus the solution to the eigenvector problem [24]. In a formal analysis of this issue [17], differentiating the Ncuts

[1] 12/3/07 LJG: By using the code we distributed on the web, it was discovered by Lu Yu that there was a bug in our Normalized Cuts implementation. After fixing this bug, Lu Yu demonstrated that the stability of Normalized Cuts is comparable to the Isoperimetric Algorithm. The text/figures in this document are reprinted without modification from the PAMI publication. Thanks to Lu Yu for discovering this bug.



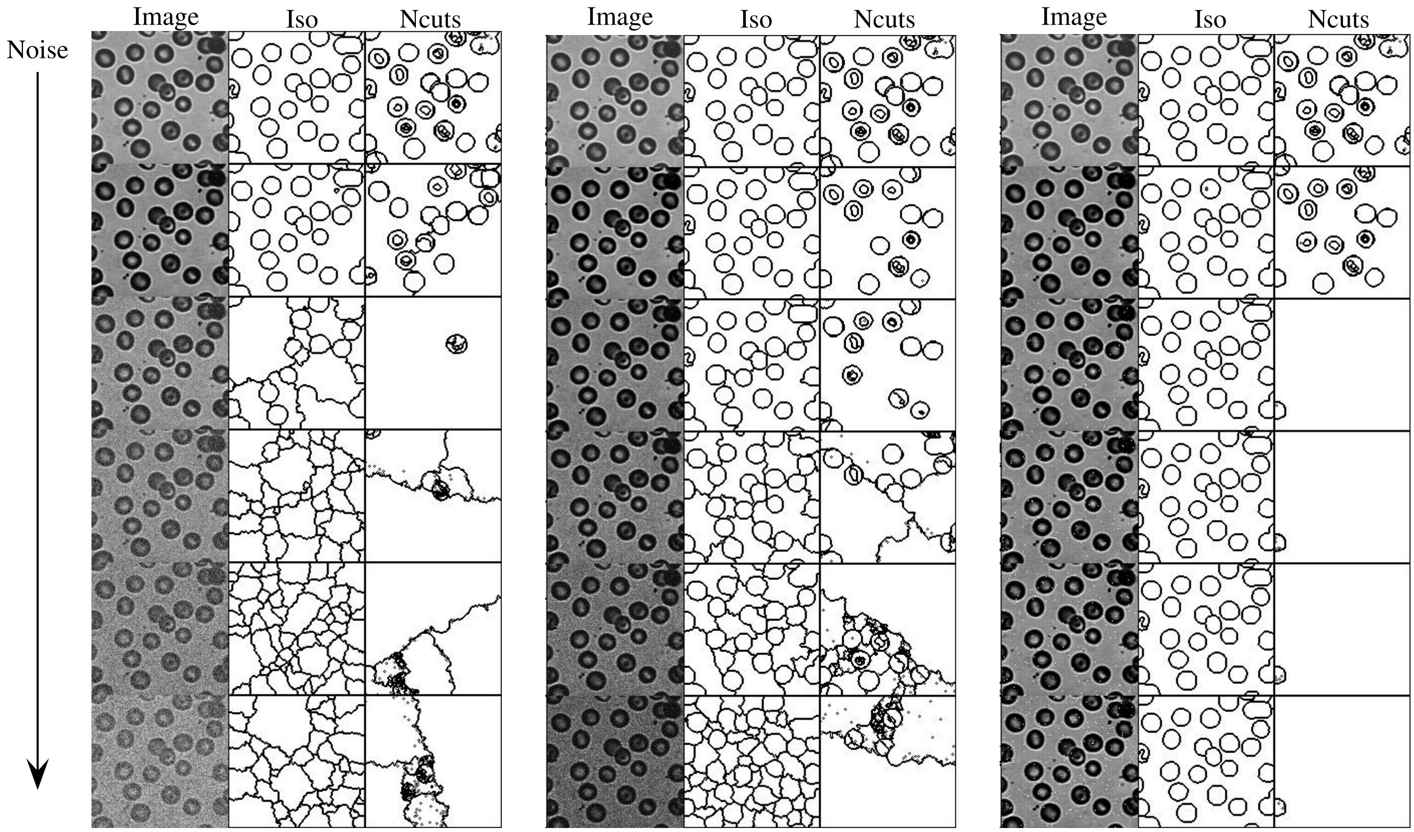
(a) Additive noise (b) Multiplicative noise (c) Shot noise

Fig. 3. Comparison of segmentation stability between the isoperimetric algorithm and Ncuts. Progressive amounts of additive, multiplicative and shot noise were applied to an artificial image of a white circle on a black background, for which the correct number of segments is exactly one. The x-axis represents an increasing noise variance for the additive and multiplicative noise, and an increasing number of "shots" for the shot noise. The y-axis indicated the number of segments found by each algorithm. The solid line represents the number of segments found by the isoperimetric algorithm and the dashed line represents the number of segments found by the Ncuts algorithm. The underlying graph topology was the 4-connected lattice with $\beta = 95$ for the isoperimetric algorithm and $\beta = 35$ for the Ncuts algorithm. Ncuts `stop` criterion = 10^{-2} (relative to the Ncuts criterion) and isoperimetric `stop` criterion = 10^{-5}. In all cases, the isoperimetric algorithm outperforms Ncuts, most dramatically in response to shot noise due to the instability of the eigenvector for multiple, nearly disconnected, regions — see text. The β and `stop` values for each algorithm were chosen empirically to produce the best results for that algorithm in response to noise.

equation with respect to an edge weight reveals that the derivative of the solution to the Ncuts equation is highly dependent on the current Fiedler value, even taking degenerate solutions for some values. By contrast, the derivative of the isoperimetric solution has no poles. Instability in spectral methods due to algebraic multiplicity of the Fiedler value is a common problem in implementation of these algorithms. This analysis suggests that the Ncuts algorithm may be more unstable to minor changes in an image than the isoperimetric algorithm.

The relative sensitivity of Ncuts (our implementation) and the isoperimetric algorithm to noise was compared using a quantitative and a qualitative measure. Each algorithm was applied to an artificial image of a white circle on a black background, using a 4-connected lattice topology. Increasing amounts of additive, multiplicative and shot noise were applied, and the number of segments output by each algorithm was recorded. Results of this experiment are recorded in Figure 3.

In each comparison, additive, multiplicative, and shot noise were used to test the sensitivity of the two algorithms to noise. The additive noise was zero-mean Gaussian noise with variance ranging from 1–20$\%$ of the brightest luminance. Multiplicative noise was introduced by multiplying each pixel by a unit-mean Gaussian variable with the same variance range as above. Shot noise was added to the image by randomly selecting pixels that were fixed to white. The number of "shots" ranged from 10 to 1,000. Although additive and multiplicative noise heavily degrades the solution found the Ncuts algorithms, the isoperimetric algorithm degrades more gracefully. Even the presence of a significant amount of shot noise does not seriously disrupt the isoperimetric algorithm, but it significantly impacts the convergence of Ncuts to any solution. An additional, qualitative, comparison is shown in Figure 4, yielding similar results.



(a) Additive noise (b) Multiplicative noise (c) Shot noise

Fig. 4. Stability analysis relative to additive, multiplicative and shot noise. Each row represents an increasing amount of noise of the appropriate type. The top row in each subfigure is the segmentation found for the `blood1.tif` image packaged with TMMATLAB (i.e., zero noise). Each figure is divided into three columns representing the image with noise, isoperimetric segmentation and Ncuts segmentation from left to right respectively. The underlying graph topology was the 4-connected lattice for isoperimetric segmentation and an 8-connected lattice for Ncuts segmentation (due to failure to obtain quality results with a 4-connected lattice) with $\beta = 95$ for the isoperimetric algorithm and $\beta = 35$ for the Ncuts algorithm. Ncuts `stop` criterion = 5×10^{-2} (relative to the Ncuts criterion) and isoperimetric `stop` criterion = 10^{-5}. Results were slightly better for additive noise, and markedly better for multiplicative and shot noise. Note that the β and `stop` values for each algorithm were chosen empirically to produce the best results for that algorithm in response to noise. (a) Additive noise. (b) Multiplicative noise. (c) Shot noise.

C. Segmentation of natural images

Examples of the segmentation found by the isoperimetric algorithm for some natural images are displayed in Figure 5. All results in the example segmentations were obtained using the same two parameters. It should be emphasized in comparisons of segmentations produced by the Ncuts algorithm that the authors of Ncuts make use of a more fully connected neighborhood as well as fairly sophisticated spatial filtering (e.g., oriented Gabor filters at multiple scales) in order to aid in textural segmentation. The demonstrations with the isoperimetric algorithm used a basic 4-connected topology and no spatial filtering at all. Consequently, the segmentations produced by the isoperimetric algorithm should be expected to perform less well on textural cues. However, for general grayscale images, it appears to perform well, with increased numerical stability and a speed advantage of more than an order of magnitude over Ncuts (based

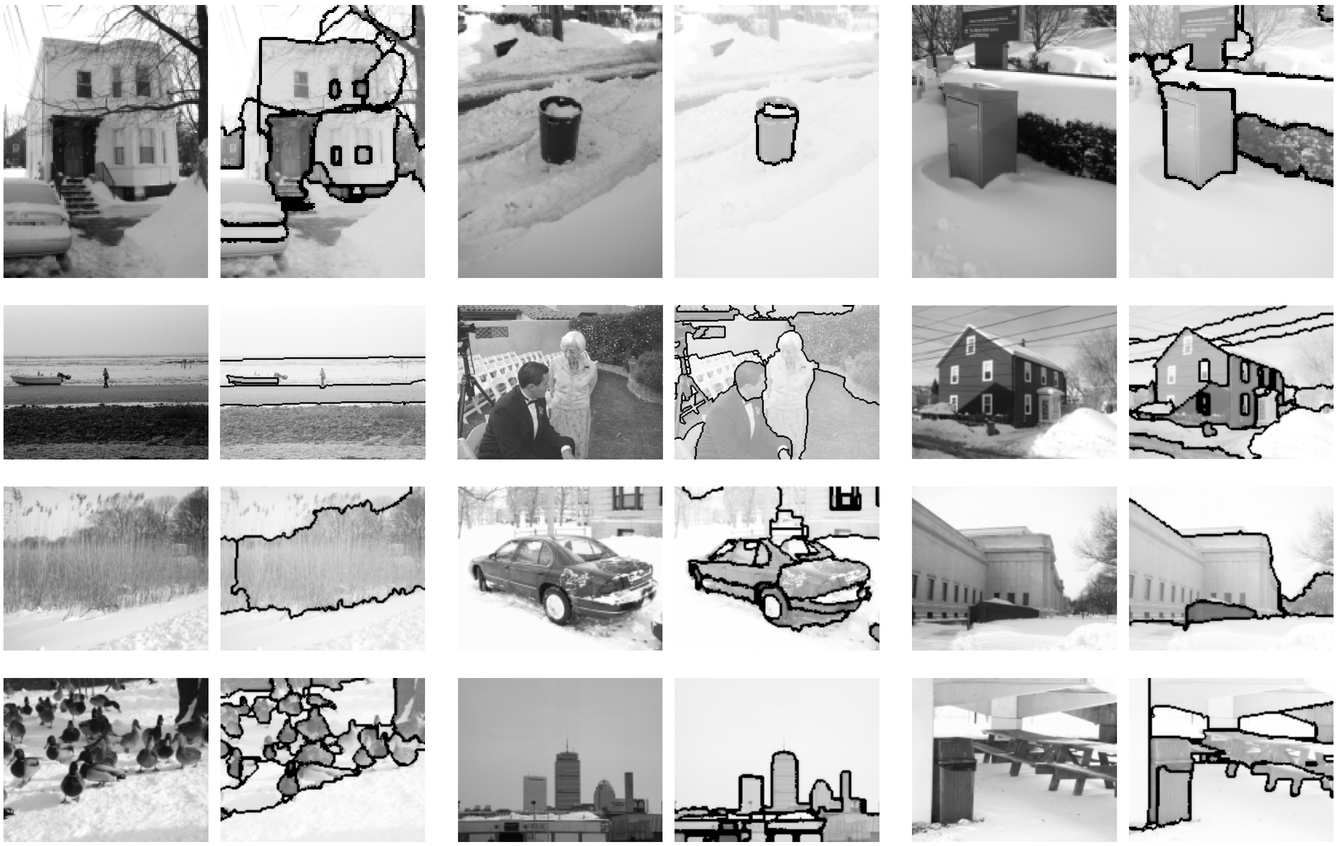


Fig. 5. Examples of unsupervised segmentations produced by the isoperimetric algorithm using the same parameters ($\beta = 95$, `stop` = 10^{-5}). Note that only intensity information was used by the algorithm (i.e., no texture or multiscale information was used). Our MATLAB implementation required approximately 10–15 seconds to segment each image. More segmentation results from the same database may be found at `http://eslab.bu.edu/publications/grady2003isoperimetric/`. Original images may be obtained from `http://eslab.bu.edu/resources/imageDB/imageDB.php/`

on our MATLAB implementation of both algorithms). Furthermore, because of the implementation (e.g., 4-connected lattice, no spatial filtering), the isoperimetric algorithm makes use of only two parameters, compared to the four basic parameters (i.e., radius, two weighting parameters and the recursion stop criterion) required by the Ncuts algorithm [4].

The asymptotic (formal) time complexity of Ncuts is roughly the same as the isoperimetric algorithm. Both algorithms have an initial stage in which nodal values are computed that requires approximately $\mathcal{O}(n)$ operations (i.e., via Lanczos or conjugate gradient). Generation of the nodal values is followed in both algorithms by an identical cutting operation. Using the MATLAB sparse matrix solver for the linear system required by the isoperimetric algorithm and the Lanczos method MATLAB employs ARPACK [31] for this calculation) to solve the eigenvalue problem required by Ncuts, the time was compared for a 10000×10000 L matrix (i.e., a 100×100 pixel image) representing a 4-connected graph (for both algorithms). Since other aspects of the algorithms are the same (e.g., generating weights from the image, cutting the indicator vector, etc.), and because solving for the indicator vector is the main computational hurdle, we only compare the time required to solve for the indicator vector. On a 1.4GHz AMD Athlon with 512K RAM, the time required to approximate the Fiedler vector for NCuts was 7.1922 seconds while application of the direct solver to the isoperimetric partitioning (13) required 0.5863 seconds. In terms of actual computation time (using MATLAB), this result demonstrates that solving the central computation for the isoperimetric algorithm is more than an order of magnitude faster than solving the central computation required by the Ncuts algorithm.

IV. Conclusion

We have recently described a new algorithm for graph partitioning that attempts to find sets with a low isoperimetric ratio [32]. Here, this algorithm is applied to the problem of image segmentation, and application of it for data clustering was additionally explored in [17]. The algorithm was compared with Ncuts to demonstrate that it is faster and more stable, while providing visually comparable results with less pre-processing. The isoperimetric algorithm additionally admits interpretation in terms of circuit theory, random walks and combinatorial PDEs, lending the depth of these well-researched literatures to analysis of the algorithm's behavior. The (MATLAB) code used to generate all the figures in this paper will be available upon publication at `http://eslab.bu.edu/publications/` using the Graph Analysis Toolbox available at `http://eslab.bu.edu/software/graphanalysis/`.

Developing algorithms to process a distribution of data on graphs is an exciting area. Many biological sensory units are non-uniformly distributed in space (e.g., vision, somatic sense) with spatial distribution often differing radically between species [33]. The ability to develop algorithms that allow an arbitrary choice for the distribution of sensors (or data of any sort) represents a large step over existing algorithms that require a regular, shift-invariant lattice. For modeling image processing on biological space-variant systems, we have found this property to be a necessity.

Since the graph representation of an image is not tied to any notion of dimension, the algorithm applies equally to graph-based problems in N-dimensions as it does to problems in two dimensions. Suggestions for future work are applications to segmentation in space-variant architectures, supervised or unsupervised learning, 3-dimensional segmentation, and the segmentation/clustering of other areas that can be naturally modeled with graphs.

References

[1] Jefferey Cheeger, "A lower bound for the smallest eigenvalue of the Laplacian," in *Problems in Analysis*, R.C. Gunning, Ed., pp. 195–199. Princeton University Press, Princeton, NJ, 1970.

[2] Bojan Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, pp. 274–291, 1989.

[3] Bojan Mohar, "Isoperimetric inequalities, growth and the spectrum of graphs," *Linear Algebra and its Applications*, vol. 103, pp. 119–131, 1988.

[4] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[5] Sudeep Sarkar and Padmanabhan Soundararajan, “Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 504–525, May 2000.

[6] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation,” *IEEE Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1101–1113, 1993.

[7] Yuri Boykov and Marie-Pierre Jolly, “*Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images,” in *International Conference on Computer Vision*, July 2001, vol. I, pp. 105–112.

[8] Fan R. K. Chung, *Spectral Graph Theory*, Number 92 in Regional conference series in mathematics. American Mathematical Society, Providence, R.I., 1997.

[9] Jozef Dodziuk, “Difference equations, isoperimetric inequality and the transience of certain random walks,” *Transactions of the American Mathematical Soceity*, vol. 284, pp. 787–794, 1984.

[10] Jozef Dodziuk and W. S Kendall, “Combinatorial Laplacians and the isoperimetric inequality,” in *From local times to global geometry, control and physics*, K. D. Ellworthy, Ed., vol. 150 of *Pitman Research Notes in Mathematics Series*, pp. 68–74. Longman Scientific and Technical, 1986.

[11] Alex Pothen, Horst Simon, and Kang-Pu Liou, “Partitioning sparse matrices with eigenvectors of graphs,” *SIAM Journal of Matrix Analysis Applications*, vol. 11, no. 3, pp. 430–452, 1990.

[12] Y.C. Wei and C.K. Cheng, “Ratio cut partitioning for hierarchical designs,” *IEEE Trans. on CAD*, vol. 10, no. 7, pp. 911–921, July 1991.

[13] C. Hagen and A. Kahng, “New spectral methods for ratio cut partitioning and clustering,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[14] Song Wang and Jeffrey Mark Siskund, “Image segmentation with ratio cut,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675–690, June 2003.

[15] Norman Biggs, *Algebraic Graph Theory*, Number 67 in Cambridge Tracts in Mathematics. Cambridge University Press, 1974.

[16] Miroslav Fiedler, *Special matrices and their applications in numerical mathematics*, Martinus Nijhoff Publishers, 1986.

[17] Leo Grady, *Space-Variant Computer Vision: A Graph-Theoretic Approach*, Ph.D. thesis, Boston University, Boston, MA, 2004.

[18] Franklin H. Branin, Jr., “The algebraic-topological basis for network analogies and the vector calculus,” in *Generalized Networks, Proceedings*, Brooklyn, N.Y., April 1966, pp. 453–491.

[19] Peter Doyle and Laurie Snell, *Random walks and electric networks*, Number 22 in Carus mathematical monographs. Mathematical Association of America, Washington, D.C., 1984.

[20] Leo Grady and Gareth Funka-Lea, “Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials,” in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV 2004 Workshops CVAMIA and MMBIA*, Milan Šonka, Ioannis A. Kakadiaris, and Jan Kybic, Eds., Prague, Czech Republic, May 2004, number LNCS3117 in Lecture Notes in Computer Science, pp. 230–245, Springer.

[21] Prasad Tetali, “Random walks and the effective resistance of networks,” *Journal of Theoretical Probability*, vol. 4, no. 1, pp. 101–109, 1991.

[22] Daniel A. Spielman and Shang-Hua Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” Tech. Rep. UCB CSD-96-898, University of California, Berkeley, 1996.

[23] William N. Anderson, Jr. and Thomas D. Morley, “Eigenvalues of the Laplacian of a graph,” Tech. Rep. TR 71-45, University of Maryland, October 1971.

[24] Gene Golub and Charles Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition, 1996.

[25] John Gilbert, Cleve Moler, and Robert Schreiber, “Sparse matrices in MATLAB: Design and implementation,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 333–356, 1992.

[26] Miroslav Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak Mathematical Journal*, vol. 23, no. 98, pp. 298–305, 1973.

[27] Norga Alon and V.D. Milman, “λ_1, isoperimetric inequalities for graphs and superconcentrators,” *Journal of Combinatorial Theory, Series B*, vol. 38, pp. 73–88, 1985.

[28] Y.F. Hu and R.J. Blake, “Numerical experiences with partitioning of unstructured meshes,” *Parallel Computing*, vol. 20, pp. 815–829, 1994.

[29] Leo Grady and Eric L. Schwartz, "Faster graph-theoretic image processing via small-world and quadtree topologies," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, June 27 – July 2 2004, IEEE Computer Soceity, vol. 2 of *Conference on Computer Vision and Pattern Recognition*, pp. 360–365, IEEE.

[30] Mark Fineman, *The Nature of Visual Illusion*, Dover Publications, 1996.

[31] R. B. Lehoucq, D. C. Sorenson, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, 1998.

[32] Leo Grady and Eric L. Schwartz, "The isoperimetric algorithm for graph partitioning," *SIAM Journal on Scientific Computing*, 2005, In press. See http://cns.bu.edu/˜lgrady/grady2005isoperimetric.pdf.

[33] Austin Hughes, "The topography of vision in mammals of contrasting life style: Comparative optics and retinal organization," in *The Visual system in vertebrates*, Frederick Crescitelli, Ed., vol. 7 of *The Handbook of Sensory Physiology*, chapter 11, pp. 613–756. Springer-Verlag, Berlin, New York, 1977, Part 5.